

# IFM 25907 – Développement d'Applications Mobiles Android

## Evaluation Sommative #1

Hiver 2023

Durée: 12 heures

Travail individuel

### 1. Introduction

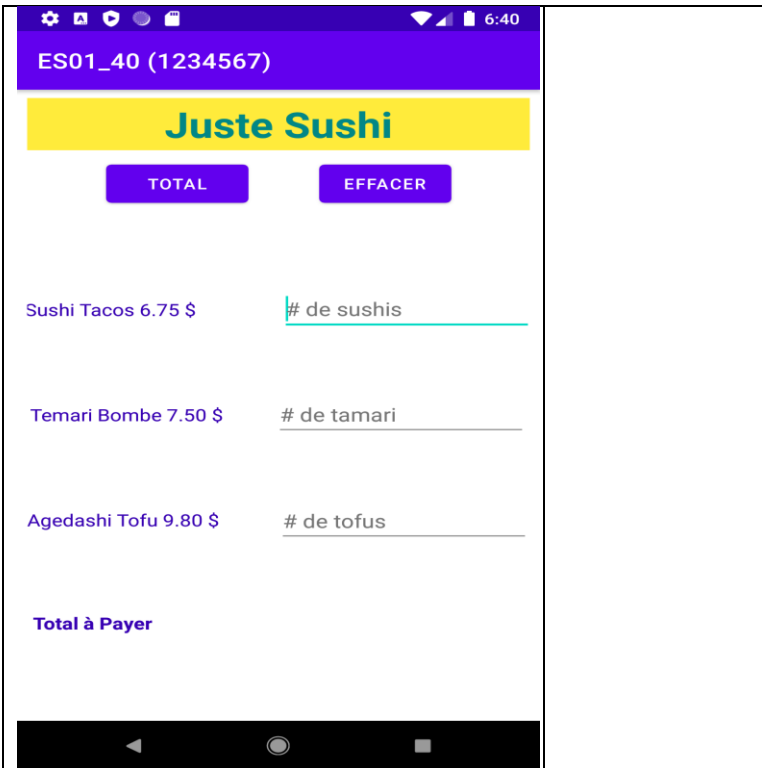
Ceci est un test pratique. Vous devez compléter le projet *Android Studio* fourni puis le soumettre via *eCité*.

Vous devez solutionner le problème en créant un projet *Android Studio* avec les fonctionnalités demandées ci-dessous, compresser ce projet (format ZIP) et le déposer dans le pigeonier à cet effet sur *eCité*.

Vous pouvez demander de l'aide de l'institutrice durant l'évaluation (celle-ci se réserve le droit de ne pas répondre à certaines de vos questions). Vous ne devez en aucun cas communiquer avec d'autres personnes que l'institutrice.

### 2. Description

Dans le cadre de cette question, vous devez réaliser un projet *Android Studio* afin d'implanter une application de commande de plats de restaurant. L'utilisateur saisit le nombre de plats désiré. L'application calcule le total à payer. L'utilisateur peut effacer le tout et recommencer l'opération.



L'application telle qu'elle devrait apparaître sur le téléphone.

### 3. Spécifications

Voici les spécifications à implanter:

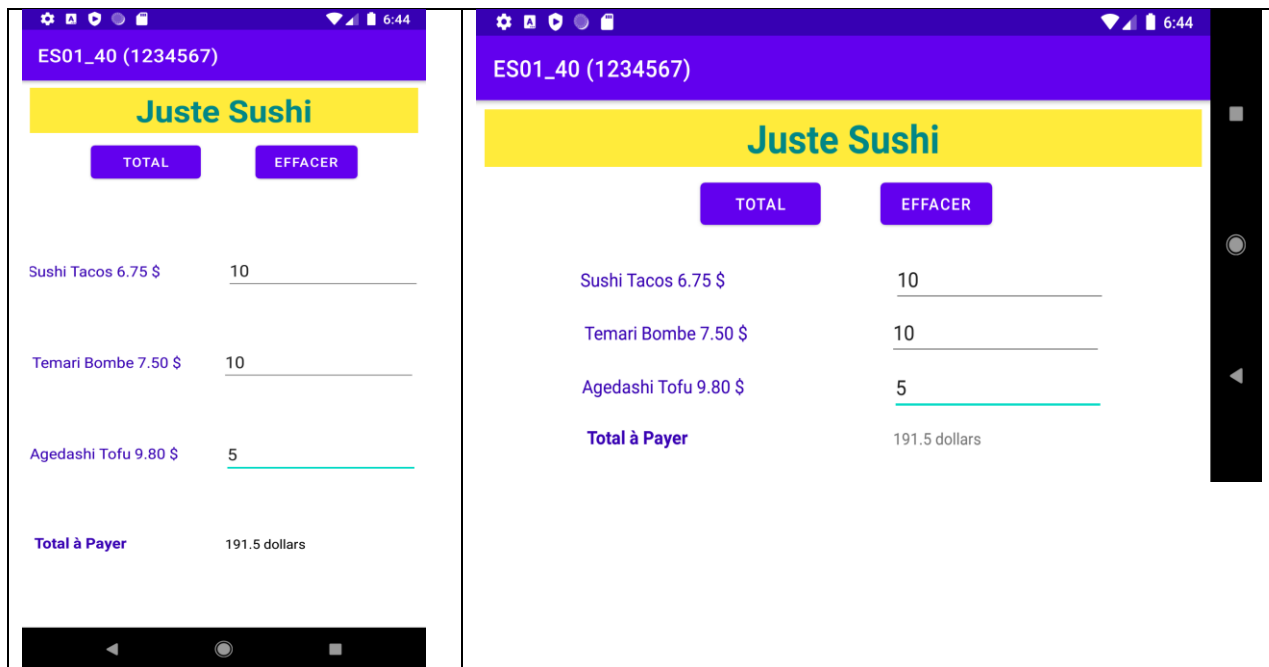
1. Le nom de l'application est '**ES01\_40**' plus votre matricule
2. Le barre titre de l'application doit afficher votre matricule étudiant à la place de **1234567**.
3. L'application doit afficher le titre '**Juste Sushi**' vec le formatage nécessaire.
4. Les 4 champs sur la gauche sont des TextViews
5. Les 3 premiers champs sur la droite sont de type EditText qui acceptent uniquement des nombres entiers.
6. Le dernier champ est un Textview qui affiche le total de la commande et les messages d'erreur.
7. '**Effacer**' et '**Total**' sont des boutons.
8. Les erreurs sont affichées selon les condition décrites à la section 4 ci-dessous.
9. Lorsque l'utilisateur entre le nombre de plats désiré, il peut ou bien effacer avec le bouton **Effacer** ou appuyer sur '**Total**'

10. Si l'utilisateur clique sur **Effacer**, tous les champs EditText et le champ **Total** doivent être effacés
11. Si l'utilisateur appuie sur '**Total**', le total de la commande doit être affiché dans le champ **Total** selon cette formule:
  - i. Le prix d'un Sushi Tacos est de 6.75\$
  - ii. Le prix d'un Temari Bombe est de 7.50\$
  - iii. Le prix d'un Agedashi Tofu est de 9.80\$


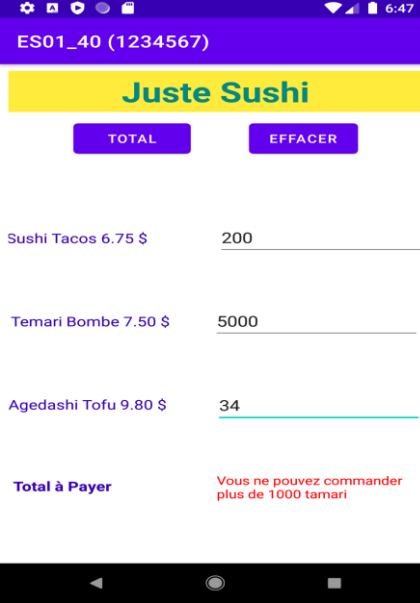

Exemples:

Voir la vidéo jointe avec ce document.

12. La disposition des éléments de l'interface doit s'adapter de façon appropriée à une réorientation de l'appareil.
13. Les valeurs entrées et calculées doivent être maintenues après un changement d'orientation de l'appareil.



#### 14. Exemples d'exécution

#### 4. Messages d'erreur

1. Lorsque l'utilisateur entre un chiffre supérieur à 1000 pour le nombre de sushis, le calcul ne doit pas être effectué et ce message en **rouge** doit être affiché dans le Textview :  
**Vous ne pouvez commander plus de 1000 sushis**
2. Lorsque l'utilisateur entre un chiffre supérieur à 1000 pour le nombre de temaris, le calcul ne doit pas être effectué et ce message en **rouge** doit être affiché dans le Textview : **Vous ne pouvez commander plus de 1000 temaris**
3. Lorsque l'utilisateur entre un chiffre supérieur à 1000 pour le nombre de tofus, le calcul ne doit pas être effectué et ce message en **rouge** doit être affiché dans le Textview:  
**Vous ne pouvez commander plus de 1000 tofus**
4. Lorsque l'utilisateur n'entre aucun chiffre dans aucun des champs, le calcul ne doit pas être effectué et ce message en **rouge** doit être affiché dans le Textview : **Entrez des valeurs**
5. Lorsque l'utilisateur entre des nombres non positifs dans un des 3 champs, le calcul ne doit pas être effectué et ce message en **rouge** doit être affiché dans le Textview : **Les nombres doivent être positifs**
6. Lorsque l'utilisateur clique sur '**Effacer**', tous les champs doivent être effacés, y compris le message d'erreur.
7. Toutes les chaînes de caractères du Layout Constraint doivent être déclarées dans le fichier '**strings.xml**'.
8. Votre fichier **MainActivity.java** doit contenir cette information en haut du fichier:

```
/**  
 * Nom et Prénom:  
 *  
 * Matricule:  
 *  
 * Groupe : 040  
 *  
 * Examen : ES01  
 *  
 * Date :
```

\*/

**Note:**

*Veuillez remplir les champs avec l'information requise.*

## 5. Suggestions

1. Toutes les notions requises pour réaliser ce projet ont été étudiées en classe et mises en pratique durant les évaluations formatives.
2. Vous devez intégrer votre matricule étudiant dans la barre titre de l'application en modifiant le fichier *res/values/strings.xml* du projet.
3. Définir des constantes Java lorsque nécessaire
4. Essayez d'intégrer les messages d'erreur dans **strings.xml**
5. Ajouter assez de commentaires pour expliquer ce que votre code fait.

## 6. Avertissement

1. Votre projet doit être compilable. S'il ne compile pas, un zéro est automatique.
2. L'application doit démarrer sans erreur, c-à-d, pas d'exception au démarrage.

## 7. Procédure pour obtenir l'archive du projet Android

1. Cliquer sur File
2. Sélectionner **Export**
3. Cliquer sur **Export to Zip File ...**
4. Donnez ce nom au fichier Zip:

ES01\_20\_VotreMatricule.zip

Par exemple, Pierre Joly avec le matricule 1234567 va sauvegarder ainsi:

ES01\_20\_1234567.zip

5. Android Studio vous demandera de choisir l'endroit où vous voulez sauvegarder votre archive
6. Soumettre ce fichier zip sur eCité

Une fois votre projet complété, n'oubliez pas de le soumettre dans le pigeonier à cet effet via *eCité*.